

REMARKS

Applicant is in receipt of the Office Action mailed July 25, 2005. Claims 1-4, 6, 12, 15-16, 19-21, 23-25, 32-44, 49-50, 53-60, and 66-71 have been amended. Claims 26-30 have been canceled. New claims 72-89 have been added. Claims 1-25, 32-44, 46-60, and 62-89 are currently pending in the application. Reconsideration of the present case is earnestly requested in light of the following remarks.

Allowable Subject Matter

Claims 16-18 and 71 were objected to as being dependent upon a rejected base claim, but the Examiner indicated that these claims would be allowable if re-written in independent form. Applicant respectfully thanks the Examiner for consideration of these claims. New independent claim 87 has been added which substantially incorporates the limitations of claim 71. Applicant thus submits that claim 87 and its dependent claims 88 and 89 are in condition for allowance.

Section 102 Rejections

Claims 1-7, 10-15, 19-30, 32-44, 46-60, and 62-70 were rejected under 35 U.S.C. 102(e) as being anticipated by Bailey et al., U.S. Patent No. 6,684,385 (hereinafter "Bailey"). Applicant respectfully traverses this rejection.

Taking amended claim 36 as an exemplary claim, the claim recites as follows:

1. (Currently Amended) A computer-implemented method for creating a graphical program, the method comprising:

creating a graphical user interface for the graphical program in response to first user input;

displaying a first node for receiving user interface events in a block diagram for the graphical program in response to second user input;

receiving third user input explicitly specifying one or more user interface events to configure for the first node;

configuring the first node to receive the one or more user interface events explicitly specified by the third user input during execution of the graphical program; and

associating one or more portions of graphical code with the first node in response to fourth user input, wherein each portion of graphical code comprises one or more nodes for responding to one or more of the user interface events which the first node is configured to receive.

Applicant notes that the features recited in claim 1 allow a user (developer) to configure a graphical program to receive and respond to specific user interface events of the user's choosing, i.e., the one or more user interface events explicitly specified by the third user input. The program development environment taught in Bailey does not provide its users with this capability.

To develop an application program in Bailey's system, the developer (user) selects icons for performing various tasks. In response, the program development environment places corresponding symbols in the designer window. The developer then graphically links these symbolic representations by drawing wires between them in order to create a data and/or execution control flow diagram. (Col. 9, lines 45-54)

Bailey teaches that the program development environment automatically generates various event-handler procedures as the user draws wires between the symbolic representations displayed in the designer window (Col. 12, line 60 – Col. 13 line 15; and Col. 13, line 58 – Col. 14 line 14). "Each symbolic representation in designer window 406 preferably includes one or more terminals disposed about it. These terminals, moreover, are associated with some pre-defined combination of the properties, methods and/or events of the respective program object that is symbolically represented." (Col. 10, line 65 – Col. 11, line 5). The event-handlers generated by the program development environment are operable to perform various functions, depending on which symbolic representations are linked and depending on which terminals on the symbolic representations the developer connects the wires to.

For example, Bailey describes an example in which the developer draws a wire between a terminal 430c of a vertical scroll bar program object and a symbolic representation 434 of a label program object. In this example, the terminal 430c is associated with both the DataReady event and the Value property of the respective vertical scroll bar program object. Thus, in response to the developer drawing the wire, the program development environment automatically creates event handler program code that sets the label object's Caption property to the value of the vertical scroll bar object's Value property when the vertical scroll bar object's DataReady event occurs. (Col. 13, line 15 – Col. 14, line 14).

Thus, Bailey teaches a program development environment that essentially abstracts the user (developer) from the details of event-driven programming by automatically creating event-handlers on the user's behalf. In contrast, rather than abstracting the user from these details, claim 1 recites the element of, "receiving third user input explicitly specifying one or more user interface events to configure for the first node". Bailey's system fails to provide users with the ability to explicitly specify particular user interface events. Instead, the user simply connects various terminals of symbols displayed in the designer window, and the program development environment automatically generates event-handlers that reflect these connections. As described above, the terminals of a symbol are associated with some pre-defined combination of the properties, methods and/or events of the respective program object. Thus, the event-handler is generated based on which properties, methods, and events are associated with the terminals to which the user connected the wires. The user does not explicitly specify any particular user interface events to which he wants the program to respond.

For example, in the example described above, an event-handler associated with the vertical scroll bar object's DataReady event is automatically generated. However, the user does not explicitly specify the DataReady event. Instead, the DataReady event is indirectly (non-explicitly) specified because it is associated with the terminal 430c to which the user connected the wire.

Bailey also fails to teach the element of, "displaying a first node for receiving user interface events in a block diagram for the graphical program in response to second user input." As described above, Bailey's system attempts to abstracts the user (developer) from the details of event-driven programming. Thus, there would be no reason for Bailey's system to provide users with access to a node for responding to user interface events, where the node can be configured to receive one or more specific user interface events explicitly specified by the user.

Applicant also notes that Bailey describes two ways of creating event handlers: 1) automatically generating Visual Basic code and adding it to the application program; and 2) automatically instantiating a "wire program object" in the form window for providing event handler functionality for other program objects in the form window (Col. 13, line 64 – Col. 14, line 14). Both of these techniques involve automatically adding an event-

handler to the program “behind-the-scenes”, so to speak. In other words, the user may not even be aware that the event-handler was added to the program. In contrast, claim 1 recites that the first node for receiving user interface events is displayed in the block diagram in response to user input.

Furthermore, Bailey fails to teach the element of, “associating one or more portions of graphical code with the first node in response to fourth user input, wherein each portion of graphical code comprises one or more nodes for responding to one or more of the user interface events which the first node is configured to receive.” As described above, Bailey teaches automatically generating an event handler. Bailey does not teach receiving user input to associate a portion of graphical code with a first node, where the portion of graphical code comprises one or more nodes for responding to a user interface event which the first node is configured to receive.

Thus, for at least the reasons set forth above, Bailey does not teach the features recited in claim 1. Applicant thus submits that claim 1, and claims dependent thereon, are patentable over Bailey. Inasmuch as the other independent claims recite similar features as those discussed above with respect to claim 1, Applicant submits that the other independent claims, and claims dependent thereon, are patentable over Bailey.

Applicant also submits that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the independent claims have been shown to be patentably distinct, a further discussion of the dependent claims is not necessary at this time.

Section 103 Rejections

Claims 8-9 were rejected under 35 U.S.C. 103(a) as being unpatentable over Bailey and Zizzo (U.S. Pat. No. 6,578,174). Applicant respectfully traverses this rejection.

Applicant respectfully submits that there is no teaching, suggestion, or motivation to combine Bailey and Zizzo in either of the references or in the prior art. As held by the U.S. Court of Appeals for the Federal Circuit in *Ecolchem Inc. v. Southern California Edison Co.*, an obviousness claim that lacks evidence of a suggestion or motivation for

one of skill in the art to combine prior art references to produce the claimed invention is defective as hindsight analysis.

Furthermore, the showing of a suggestion, teaching, or motivation to combine prior teachings “must be clear and particular. . .Broad conclusory statements regarding the teaching of multiple references, standing alone, are not ‘evidence’.” *In re Dembiczak*, 175 F.3d 994, 50 USPQ2d 1614 (Fed. Cir. 1999). The art must fairly teach or suggest to one to make the specific combination as claimed. That one achieves an improved result by making such a combination is no more than hindsight without an initial suggestion to make the combination.

Applicant respectfully submits that there is no clear teaching or suggestion for combining Bailey and Zizzo and notes that the Office Action does not provide evidence of such a teaching or suggestion.

Furthermore, Applicant respectfully submits that it is nonobvious to combine Bailey and Zizzo, and that even if Bailey and Zizzo were combinable, which Applicant argues they are not, the resultant combination would still not produce the combination of elements recited in claims 8-9.

CONCLUSION



Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert & Goetzel PC Deposit Account No. 50-1505/5150-58700/JCH.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard

Respectfully submitted,

A handwritten signature in cursive script, appearing to read "JCH", written over a horizontal line.

Jeffrey C. Hood
Reg. No. 35,198
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: 10/25/2005 JCH/JLB